Confusion Matrices

Suppose that we have trained a simple binary classification model: Given an image, this model will indicate whether it is a picture of a cat or a picture of a dog. How can we evaluate our model's performance? What is a good metric for doing so?

Let us first consider what it means for the model to perform well. If the model tells us an image has a dog in it *and that image actually has a dog*, we would say it performs well. And similarly, if it says that the image has a cat *and it actually has a cat* that would also be good.

To help us think about the problem, we can construct a table that shows all of the possibilities:

		Actual class	
		Cat	Dog
Predicted class	Cat	**Correct Cat**	Incorrect Cat
	Dog	Incorrect Dog	**Correct Dog**

As you can see, the columns here represent the *actual class*—that is, whether an image *actually* has a dog or a cat. The rows represent the *predicted class*—that is, whether the model concludes that an image has a dog or a cat. When the predicted class matches the actual class (e.g., the model says the image has a cat and the image does indeed have a cat), this is a *correct* classification.

QUESTION 1 OF 3

We can use a table like this to organize our model's output. Suppose that we give the classifier test data containing 10 images, and the results are as follows:

		Actual class	
		Cat	Dog
Predicted class	Cat	**5**	1
	Dog	1	**3**

How well has the model performed?

○ The model performed perfectly—there were no misclassified images at all

✓ The model performed well, though not perfectly—there were a few misclassified images

○ The model performed poorly—most of the images were misclassified

○ The model performed very badly—it misclassified *all* of the images

SUBMIT

QUESTION 2 OF 3

OK, here is a different set of results:

		Actual class	
		Cat	Dog
Predicted class	Cat	**0**	4
	Dog	6	**0**

How well has the model performed this time?

○ The model performed perfectly—there were no misclassified images at all

○ The model performed quite well, though not perfectly—there were a few misclassified images

○ The model performed poorly—most of the images were misclassified

✓ The model performed very badly—it misclassified *all* of the images

SUBMIT

The key is to look at the diagonals. If the upper left and lower right cells are high relative to the others, then the model is making more correct classifications than incorrect classifications:

		Actual class	
		Cat	Dog
Predicted class	Cat	**4**	1
	Dog	2	**3**

Whereas if the upper right and lower left cells are comparatively higher, the model is making more incorrect classifications:

		Actual class	
		Cat	Dog
Predicted class	Cat	2	**3**
	Dog	**4**	1

This type of table is called a **confusion matrix**. A confusion matrix gets its name from the fact that it is easy to see whether the model is getting *confused* and misclassifying the data.

You will often see the confusion matrix represented in a more general, abstract form that uses the terms *positive* and *negative*:

		Actual class	
		Positive	Negative
Predicted class	Positive	**True Positives (TP)**	False Positives (FP)
	Negative	False Negatives (FN)	**True Negatives (TN)**

- **True positives** are the *positive* cases that are *correctly* predicted as *positive* by the model
- **False positives** are the *negative* cases that are *incorrectly* predicted as *positive* by the model
- **True negatives** are the *negative* cases that are *correctly* predicted as *negative* by the model
- **False negatives** are the *positive* cases that are *incorrectly* predicted as *negative* by the model

QUESTION 3 OF 3

Which one of the following is **incorrect** about confusion matrices?

○ The sum of TN and FN tells us the number of cases predicted as negative by the model

✓ The sum FP and TP tells us the number of actual positive cases in the dataset

○ The sum of FP and TN tells us the number of actual negative cases in the datasets

○ The sum of TP and TN tells us the number of cases correctly predicted by the model

SUBMIT

We can construct several different very useful metrics from a confusion matrix—and that's what we'll look at next.

NEXT